de modo que nenhum vértice tenha a mesma cor que um adjacente

Coloração → atribuição de uma cor (ou label) p/ cada vértice. Nº cromático = mínimo de cores necessárias p/ colorir

Algoritmo parcial: ordena vértices em ordem decrescente de graus e, para cada um deles, atribui a menor cor com a qual não tem adjacência

Alteração Estrutural → transforma o grafo em um grafo completo K_n. O menor n dos grafos gerados é o nº cromático. P/ cada grafo usa a operação α (acrescenta aresta entre 2 vértices não adjacentes) e β (condensa os 2 vértices, herdando as arestas de ambos).

Rede → possui um vértice S com apenas arestas de saída e um T com apenas arestas de entrada. Tirando esses 2, entrada deve ser igual a saída. Fluxo da rede é tudo que sai da origem e capacidade é tudo que chega em T.

Ford-Fulkerson → calcula ~~o~~ fluxo máximo em uma rede

↳ cria uma rede residual, com arestas que invertem os conceitos:

aresta direta: quando capacidade > fluxo, ainda há fluxo disponível de V p/ U. O valor da aresta é a diferença entre capacidade e fluxo

aresta contrária: quando fluxo > 0, pode retornar (devolver) o fluxo previamente alocado, gerando aresta de U p/ V com valor igual ao fluxo

Se há um caminho de S a T na rede residual, pode aumentar o fluxo de todo o caminho na rede original em X (sendo X o menor valor no caminho)

Gera uma nova rede residual e repete até não houver mais caminhos de S a T.

Produtor X Consumidor → vértices que recebem de S são produtores (produzindo o valor da aresta (S,V)) e os que incidem em T são consumidores (consumindo o valor da aresta (V,T). Demais vértices e arestas fazem parte do caminho de transporte que pode limitar o transporte. Quando um vértice possui restrição, divide em 2 vértices V' e V'', com uma aresta (V', V'') com valor igual à restrição

Usa Ford-Fulkerson depois

Cobertura de arestas → conjunto de arestas adjacente a todos os vértices de G

" de vértices → conjunto de vértices adjacente a todas as arestas de G

Vértice dominante → " de vért. em que todo vértice de G é adjacente a algum vértice do conj. ou faz parte do próprio conj.

Associação Máxima → emparceira dois vértices, formando uma rede e usando Ford-Fulker son se encontra a associação máxima. Hopcroft-Karp também funciona, associando um vértice de um lado com outro do outro até todos terem um par. Em grafos não bipartidos usa caminho alternante

PERT-CPM → vértices representam a conclusão de tarefas, arestas as tarefas (valor da aresta é a duração)
Caminho crítico são as atividades obrigatórias de terminar em tempo p/ ñ atrasar o projeto.
data mais cedo → momento mais cedo que pode ser concluída a atividade. A data mais cedo da última atividade é a duração do projeto. Data mais tarde começa do fim e vai p/ o começo.
Quando se escolhe de duas fontes: DMC → Maior / DMT → Menor. Caminho crítico são os vértices que DMC == DMT

Ordenação Topológica → ordena o grafo deixando predecessores antes dos antecessores. Remove os vértices de grau 0 e suas arestas e o adiciona ao caminho, ou DFS que adiciona ao caminho na volta da recursão (resulta invertido)
Componentes Fortemente Conexos → pode chegar em qualquer vértice a partir de qualquer vértice. Kosaraju: DFS pós ordem para numerar os vértices. Gera G' com arestas invertidas e realiza DFS em G' começando pelo último a ser adicionado no caminho na 1ª DFS. Cada DFS resulta em uma componente fortemente conexa (repete a 2ª DFS até todo vértice ser parte de 1 componente

Caixeiro Viajante → programação dinâmica. Lança recursivamente ponto de partida e todos que devem ser visitados, salvando os caminhos e mantendo o menor.

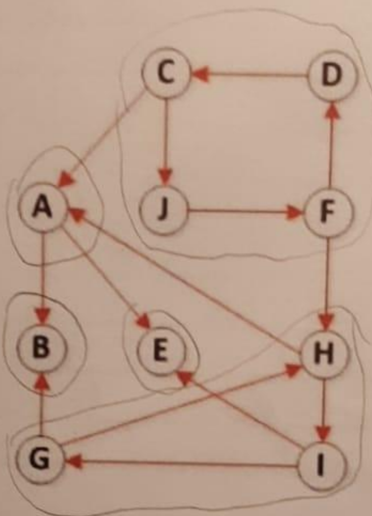
$$F(S,\{a,\dots,T\}) \to F(a,\{b,\dots,T\}) + G_{S,a} \quad \cdots \quad F(T,\{\}) + G_{a,T} \to G_{T,S}$$

→ em casos com dados excludentes, modela o grafo apenas com as arestas que geram a exceção

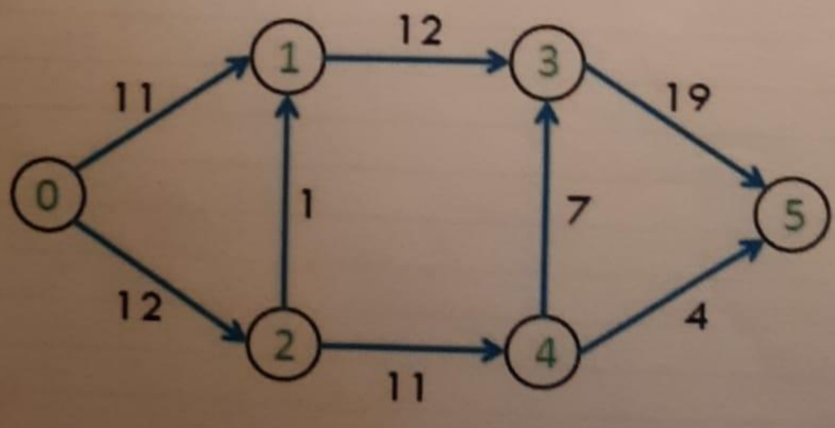
Universidade de Caxias do Sul
Área de Ciências Exatas e da Tecnologia
Teoria de Grafos – Prof. Ricardo Dorneles
Área 2 – 28/06/23

1) (2.0 pontos) Faça uma função que receba uma matriz de adjacências G[10][10] de um grafo não dirigido e verifique se o conjunto de arestas nele contido representa uma associação perfeita, ou seja, um conjunto de arestas independentes que cobre todos os vértices.

2) (1.5 pontos) Identique as componentes fortemente conexas do grafo a seguir. :



3) (1.5 pontos) Considere a rede abaixo, em que o número em cada aresta representa sua capacidade. A partir de um fluxo zerado, mostre a rede residual nas duas primeiras iterações do algoritmo de Ford Fulkerson de cálculo de fluxo máximo:



5.0

___/___/___

duardo e Teresa

```
1) int assoc_perf(int G[][]){
      int parceiros[10];
      for (int i=0; i<10; i++)
         parceiros[i]=0;

      for (int i=0; i<10; i++)
         for(int j=0; j<10; j++)
            if(G[i][j]!=-1)
               parceiros[i]+=1; // como é não dirigido, o grau sempre
                                // deve ser igual a 1, caso contrário
                                // mais de um parti ou menos de um p
      int resultado=1;
      for(int i=0; i<10; i++)   // se for 0, significa que nenhuma ar
                                //                     cobre o vér
         if(parceiros[i]!=1)    // se for >1, significa que a aresta possui ar
         { resultado=0;         // adjacentes e logo não é independent
            break; }

      return resultado;         // 0 -> não é uma associação perfeita
                                // 1 -> é uma associação perfeita
}
```

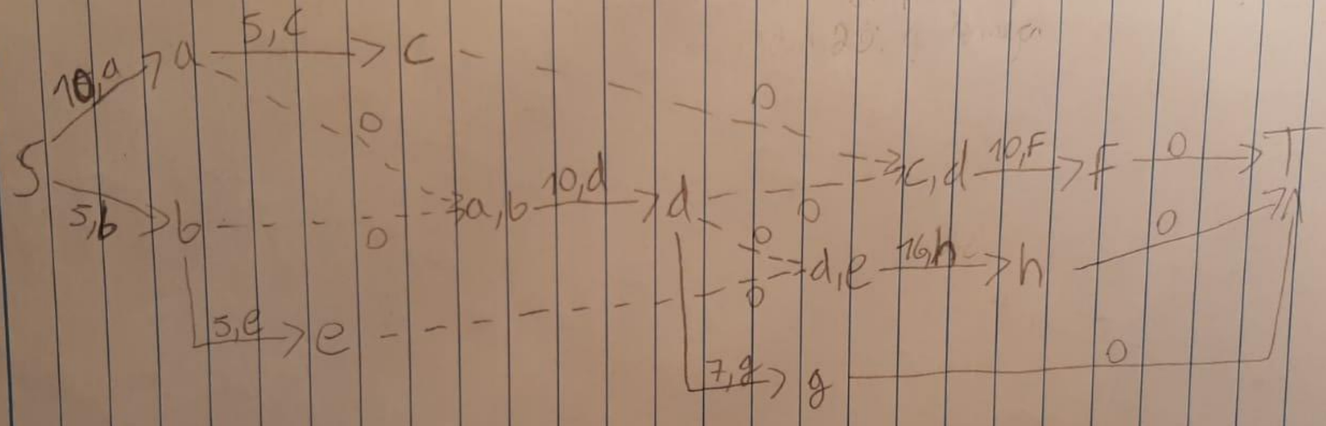
2.0

2) arestas = tarefa

vértices = conclusão da tarefa

folga = mais_tarde - mais_cedo

1.5



S	a	b	c	a,b	d	e	c,d	f	g	d,e	h	T	
0	10	5	15	10	20	10	20	30	27	20	36	36	data_mais_cedo
0	10	10	26	10	20	20	26	36	36	20	36	36	data_mais_tarde
0	0	5	11	0	0	10	6	6	9	0	0	0	folga

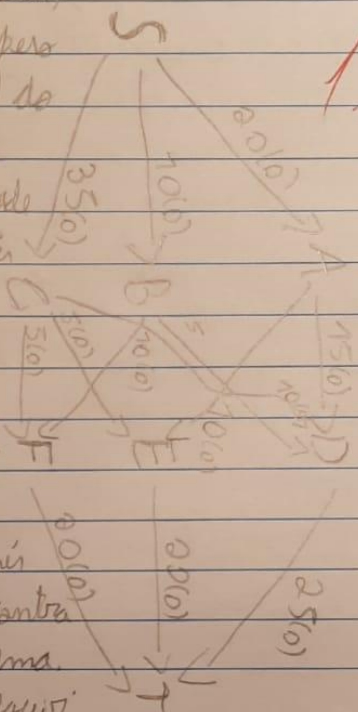
Tempo mínimo: 36

Caminho crítico: a, d, h

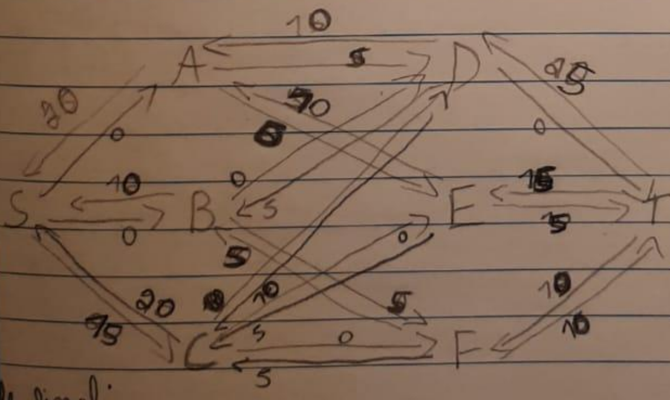
3)

1.5

-Gera-se uma rede com um vértice inicial, que possui arestas para A, B e C, com peso de arestas igual a quantidade disponível do produto. A, B e C terão arestas para D, E e F com peso igual a disponibilidade de transporte, respeitando as informações na tabela. D, E e F possuem arestas para um vértice final, com peso igual a quantidade de produto requisitada. O grafo gerado é este ao lado: →

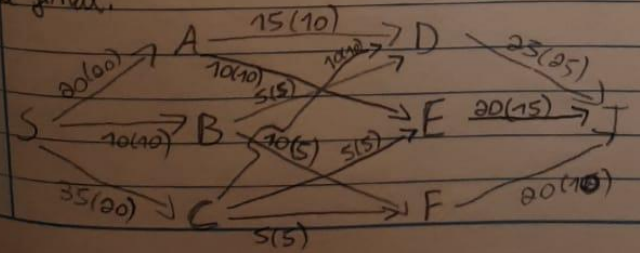


Depois, gera a rede residual e, através do algoritmo Ford-Fulkerson, se encontra a melhor resolução para o problema. A rede residual final é esta a seguir:



S-A-D-T → 15
S-A-E-T → 5
S-B-D-T → 5
S-B-F-T → 5
S-C-D-T → 5
S-C-E-T → 5
S-C-F-T → 5
S-C-D-A-E-T → 5

Rede final:



conclusão no verso →

___/___/___

→ no caso dessa situação, não é possível nem transportar tudo que é produzido e nem suprir toda a demanda.

Universidade de Caxias do Sul
Área de Ciências Exatas e da Tecnologia
Teoria de Grafos – Prof. Ricardo Dorneles
Área 2 - Parte 2 – 03/07/23

1) (2.0 pontos) Um conjunto dominante é um subconjunto de vértices tal que todo vértice do grafo está no conjunto ou é adjacente a um dos vértices. Escreva uma função (pode ser pseudo-código) que receba uma matriz de adjacências G[10][10] e um vetor Dom[10] representando um subconjunto de vértices de G (os vértices pertencentes ao subconjunto estão marcados com 1 no vetor, os que não pertencem estão marcados com 0) e verifique se o subconjunto descrito em Dom[10] é um Conjunto Dominante de Vértices, retornando: 1 - Se Dom é um conjunto dominante; 0 - Se Dom não é um conjunto dominante.

2) (1.5 pontos) Construa um diagrama PERT da tabela de tarefas a seguir. Calcule também o tempo mínimo para completá-lo, as ATIVIDADES do caminho crítico e, para cada evento, a data mais cedo, a data mais tarde e a folga.

Tarefa	Predecessoras	Duração
a	Nenhum	10
b	Nenhum	5
c	a	5
d	a,b	10
e	b	5
f	c,d	10
g	d	7
h	d,e	16

3) (1.5 pontos) De três depósitos A, B e C, dispondo respectivamente de 20,10 e 35 toneladas de um dado produto, pretende-se fazer chegar a três destinos D, E e F, respectivamente 25, 20 e 20 toneladas do produto. As disponibilidades de transporte em caminhão entre os difentes pontos são os seguintes:

	D	E	F
A	15	10	-
B	5	-	10
C	10	5	5

Mostre como a teoria de grafos pode ser utilizada para estabelecer o melhor plano de transportes, e encontre a solução para a situação específica descrita nessa questão.

* Não dirigido Eduardo E. Pereira

___/___/___ 5.0

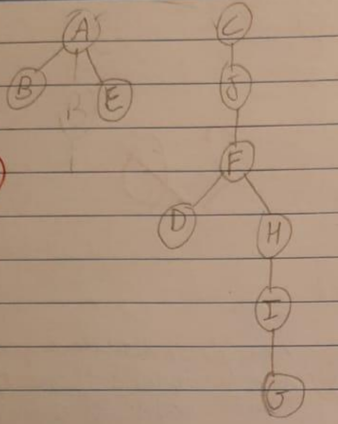
1)
```
int dominante(int G[10][10], int Dom[10]){
    int conj[10] = {0};

    for(int i=0; i<10; i++){
        if(Dom[i]==1){
            conj[i] = 1;
            for(int j=0; j<10; j++)
                if(G[i][j]==1)
                    conj[j]=1;
        }
    }

    for(int i=0; i<10; i++)
        if(conj[i]==0)
            return 0;

    return 1;
}
```

2.0

```
         1 2 3 4 5 6 7 8 9 10
DOM  = { 0 0 1 0 1 0 0 1 0 0 }
conj = { 1 1 1 1 1 1 1 1 1 1 }
```

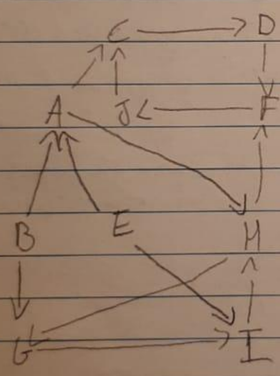
B-E-A-D-G-I-H-F-J-C

2)



1.5

G':



C-D-F-J

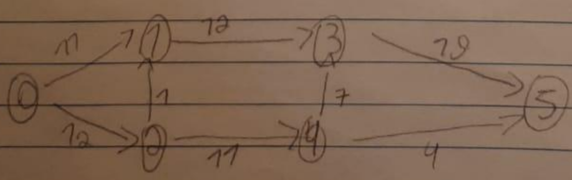
H-G-I

A

E

B

3) G' - 1ª iteração:



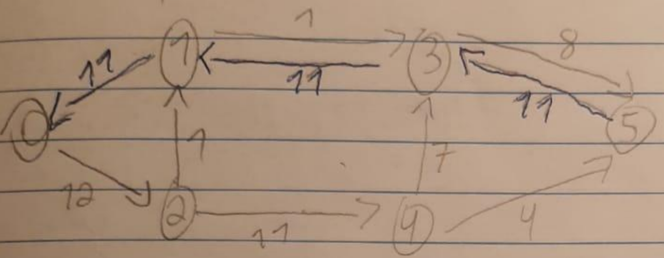
0-1-3-5-D aumenta 11

1.5

caneta: aresta contrária (fluxo>0)

lápis: aresta direta (capacidade-fluxo>0)

G' → iteração 2:



G' → iteração 3: